



Universidad de Zaragoza  
Centro Politécnico Superior  
Departamento de Ingeniería Electrónica y Comunicaciones

Proyecto de Fin de Carrera

# **Control vía ZigBee de dispositivos de iluminación con protocolo DMX para salas de estimulación multisensorial**

Autor: Daniel Domínguez Guillén  
Director: José Miguel Idiago Valero  
Ponente: Roberto José Casas Nebra

Ingeniería de Telecomunicación - Especialidad Electrónica  
Curso 2010-2011  
Diciembre 2010



*A mis padres, que sin su apoyo no  
habría sido posible llegar hasta aquí  
Gracias*

*A toda la gente de Tecnodiscap  
por estar siempre disponibles  
Y especialmente a Miguel*

*Ta Noelia y ta totz os mios amigos  
que nunca no fan burro falso  
quan se les amenista*



# **Control vía Zigbee de dispositivos de iluminación con protocolo DMX para salas de estimulación multisensorial**

## **RESUMEN**

Este Proyecto de Fin de Carrera se ha realizado en Tecnodiscap, grupo de investigación del I3A, dentro de la línea de investigación en salas de estimulación multisensorial. Se ha detectado en las salas actualmente disponibles una falta de posibilidades de interacción del usuario con el entorno, lo que supone una limitación en el desarrollo de las sesiones terapéuticas.

Se ha colaborado en el desarrollo de un sistema que permita la interacción entre diferentes elementos que pueden estar presentes en una sala de estimulación multisensorial. En concreto se ha realizado un controlador ZigBee de dispositivos de iluminación DMX.

Además del puro diseño hardware, se ha integrado el dispositivo en un driver ZigBee, que permite la auto-detección del dispositivo. Este driver funciona bajo el framework OSGi, que facilita el manejo e interconexión de dispositivos y servicios. De esta manera el objeto software creado por el driver podrá interactuar con otros dispositivos o servicios.

Por otro lado se han realizado dos diseños software. El primero de ellos se trata de un sencillo interfaz de control del dispositivo de iluminación elegido para este proyecto desde un ordenador. El segundo es una primera versión del gestor de interconexiones entre los sensores y los actuadores presentes en una sala de estimulación multisensorial, para lo que se ha colaborado tanto con alumnos de diseño industrial como con ingenieros de Tecnodiscap. Así, se ha conseguido crear una buena base para su futura ampliación con más dispositivos.

El objetivo principal ha sido el de añadir un elemento nuevo a las salas de estimulación multisensorial e integrarlo junto con otros dispositivos que se están desarrollando en paralelo; pero a la vez se pretende tener una visión clara del proceso de diseño de un dispositivo, desde la electrónica básica, pasando por el firmware, el funcionamiento de un driver y el control software del dispositivo, hasta llegar a la puesta en funcionamiento del mismo.



# INDICE DE CONTENIDOS

<b>CAPITULO 1: Introducción .....</b>	<b>10 -</b>
1.1.- Marco del proyecto.....	10 -
1.2.- Motivación del proyecto. Salas de estimulación multisensorial .....	10 -
1.3.- Desarrollo.....	12 -
1.4.- Objetivos .....	13 -
1.5.- Estructura de la memoria.....	14 -
1.6.- Distribución temporal del desarrollo del proyecto.....	15 -
<b>CAPITULO 2: Estado del arte .....</b>	<b>16 -</b>
<b>CAPITULO 3: Especificaciones del sistema.....</b>	<b>20 -</b>
3.1.- Esquema global del sistema.....	20 -
3.2.- Necesidades del usuario y condicionantes.....	22 -
<b>CAPITULO 4: Desarrollo Hardware .....</b>	<b>24 -</b>
4.1.- Protocolo DMX .....	24 -
4.2.- Actuador: foco de iluminación controlado por DMX.....	24 -
4.2.1.- Características .....	24 -
4.2.2.- Funcionamiento.....	25 -
4.3.- PCB del puente DMX – módulo ZigBee.....	26 -
4.3.1.- Características y diagramas de bloques .....	26 -
4.3.2.- Diseño de la PCB .....	31 -
4.4.- Firmware del puente DMX – módulo ZigBee.....	32 -
4.4.1.- Funcionamiento general .....	32 -
4.4.2.- Perfil ZigBee-Tecnodiscap .....	34 -
4.4.3.- Estructura de mensajes.....	37 -

<b>CAPITULO 5: Middleware y software .....</b>	<b>39 -</b>
5.1.- Framework OSGi y ontología OSGi4Aml .....	39 -
5.1.1.- OSGi .....	39 -
5.1.2.- OSGi4Aml .....	40 -
5.1.3.- Relación entre OSGi4Aml y Perfil ZigBee-Tecnodiscap .....	43 -
5.2.- Driver ZigBee .....	44 -
5.3.- Desarrollo software .....	46 -
5.3.1.- Colaboración con alumnos y profesores de Diseño Industrial .....	46 -
5.3.2.- Aplicación de desarrollo para el control del foco .....	47 -
5.3.3.- Aplicación de interacción entre dispositivos .....	48 -
<b>CAPITULO 6: Conclusiones .....</b>	<b>51 -</b>
6.1.- Cumplimiento de objetivos.....	51 -
6.2.- Desarrollos futuros .....	53 -
<b>ANEXO A: Estimulación multisensorial.....</b>	<b>56 -</b>
A.1.- Introducción. Concepto Snoezelen.....	56 -
A.2.- Salas de estimulación multisensorial.....	57 -
A.2.1.- La estimulación multisensorial.....	57 -
A.2.2.- Áreas de estimulación multisensorial.....	57 -
A.2.3.- La sala de estimulación multisensorial.....	59 -
A.2.2.- Materiales disponibles.....	60 -
<b>ANEXO B: Programmable System-on-Chip, PSoC.....</b>	<b>63 -</b>
B.1.- Características principales .....	63 -
B.2.- Arquitectura.....	65 -
<b>ANEXO C: Protocolo DMX .....</b>	<b>67 -</b>
C.1.- Introducción .....	67 -
C.2.- Nivel físico.....	68 -
C.3.- Capa de red.....	69 -
C.4.- Configuración y especificaciones del protocolo.....	69 -
C.5.- Ejemplos de dispositivos DMX.....	71 -



<b>ANEXO D: Protocolo ZigBee .....</b>	<b>- 74 -</b>
<i>D.1.- Introducción.....</i>	<i>- 74 -</i>
<i>D.2.- Estándar IEEE 802.15.4.....</i>	<i>- 75 -</i>
<i>D.3.- Protocolo ZigBee .....</i>	<i>- 76 -</i>
D.3.1.- Tipos de dispositivos.....	- 76 -
D.3.2.- Topologías y funcionamiento de la red .....	- 77 -
D.3.3.- Comunicaciones .....	- 78 -
D.3.4.- Mensajes de datos .....	- 79 -
D.3.5.- Seguridad.....	- 79 -
D.3.6.- Perfiles.....	- 80 -
D.3.7.- Campos de aplicación y resumen .....	- 81 -
<b>ANEXO E: ESQUEMATICOS Y PCB .....</b>	<b>- 83 -</b>
<i>E.1.- Esquemático del sistema y PCB .....</i>	<i>- 83 -</i>
<i>E.2.- Montaje físico final.....</i>	<i>- 87 -</i>
<b>ANEXO F: Presupuesto del hardware diseñado.....</b>	<b>- 88 -</b>
<i>F.1.- Descripción del presupuesto.....</i>	<i>- 88 -</i>
<i>F.2.- Presupuesto.....</i>	<i>- 89 -</i>
<b>INDICE DE FIGURAS .....</b>	<b>- 90 -</b>
<b>INDICE DE TABLAS.....</b>	<b>- 93 -</b>
<b>REFERENCIAS Y BIBLIOGRAFÍA .....</b>	<b>- 94 -</b>



# **CAPITULO 1**

## **Introducción**

### **1.1.- Marco del proyecto**

Este proyecto se ha desarrollado dentro del grupo de investigación Tecnodiscap [1], enmarcado en el Instituto de Investigación en Ingeniería de Aragón (I3A) [2], de la Universidad de Zaragoza. El objetivo del grupo Tecnodiscap es la mejora de la calidad de vida de las personas con algún tipo de discapacidad, dependencia o necesidades especiales mediante servicios basados en la aplicación de las Tecnologías de la Información y Comunicaciones (TIC).

En concreto, se ha integrado dentro de la línea de investigación y colaboración que lleva el grupo en el ámbito de las salas de estimulación multisensorial, siendo este proyecto parte de un conjunto de mejoras diseñadas para este tipo de instalaciones.

### **1.2.- Motivación del proyecto. Salas de estimulación multisensorial**

La idea de este proyecto surge como un aporte dentro del conjunto de soluciones y mejoras previstas ante las carencias observadas en las salas de estimulación multisensorial actualmente disponibles en los colegios con los que colabora el grupo Tecnodiscap.

Una explicación detallada del concepto de estimulación multisensorial y el funcionamiento y características de las salas se puede ver en el *Anexo A* de esta memoria, pero a modo de pequeño acercamiento, podemos señalar que dichas salas son unos espacios diseñados para la terapia con personas con algún tipo de deficiencia cognitiva o sensorial. Estos espacios pretenden ser un lugar amistoso en el que se cree un ambiente de seguridad y tranquilidad para el usuario, y así pueda expresar libremente sus emociones ante los estímulos que se le ofrecen.

Para las sesiones de terapia se dispone de gran variedad de materiales, tanto activos como pasivos (según emitan algún tipo de estímulo al usuario o simplemente sean una herramienta a utilizar sin generación de respuesta) Estos materiales son diseñados específicamente para estas salas, lo que conlleva una dependencia de los colegios hacia los fabricantes, tanto en la configuración de los dispositivos (que puede no ser exactamente la que se quiere) como en el precio de estos (normalmente alto)

Por otro lado, la gran mayoría de material disponible actualmente en las salas ofrece un nivel de interactividad muy bajo para el usuario. Son dispositivos que se pueden configurar de formas diferentes, pero sólo alguno de ellos responde en tiempo real a las acciones del usuario (en concreto, en las salas con las que se está trabajando, sólo la “tormenta de colores” aporta esta funcionalidad). Esto es algo importante, debido a que lo que se busca precisamente es provocar estímulos en el paciente, y si éste ve una respuesta directa a ellos, será más fácil que los asimile.

Ante esta realidad, desde Tecnodiscap se pretende diseñar una serie de dispositivos que sean de bajo coste para los colegios, que atiendan a las necesidades que transmitan los terapeutas y adecuen sus características a ellas, y que aporten un grado mayor de interactividad, creando un efecto de acción-reacción directa entre el usuario y el dispositivo.

También es importante resaltar que los dispositivos que se diseñen deben de ser de muy fácil manejo, dado que el usuario final va a ser un terapeuta, cuya mayor atención durante el transcurso de una sesión de terapia ha de ser hacia el paciente y no hacia el control de los dispositivos utilizados, y cuyo nivel de dominio de la tecnología no tiene por qué ser alto. Además, en muchos casos será el propio paciente el que queramos que controle el dispositivo, por lo que deberá de ser totalmente intuitivo.

### 1.3.- Desarrollo

Vistas las carencias y necesidades de las aulas de estimulación multisensorial, se plantea la creación de un sistema que permita reconocer acciones explícitas del usuario y como salida sea capaz de interaccionar con elementos que provoquen en el usuario una reacción emocional. Un ejemplo de lo anterior podría ser un sistema capaz de localizar al usuario dentro de la sala y en función de su posición cambiar la luz ambiental.

A su vez el sistema debería ser controlado en todo momento por el profesor que se encuentra trabajando en la sala, permitiéndole decidir en cada momento con que elementos del aula quiere trabajar y pudiendo establecer las relaciones causa efecto entre los diversos elementos presentes en el aula.

Para ello se propone la creación de un dispositivo de actuación sobre elementos escenográficos.

El elemento en cuestión se tratará de un sistema de control inalámbrico de todo tipo de dispositivos con protocolo DMX. Este protocolo es el estándar *de facto* de todos los dispositivos de iluminación escenográfica controlados remotamente existentes en el mercado profesional, por lo que el sistema se podrá utilizar con los dispositivos (focos de luz, máquinas de humo...) que se adapten mejor a cada instalación.

El control de los focos elegidos se realizará desde un ordenador central mediante un software diseñado al efecto, y el cual nos permitirá a su vez vincular ese control a cualquier dispositivo de interacción que hayamos diseñado dentro de este proyecto de mejora de las aulas de estimulación multisensorial.

Al ser un sistema inalámbrico, tiene una gran versatilidad a la hora de instalarlo en la sala, siendo necesarios únicamente un cable con conectores XLR-3 (explicados en el *Anexo C*) conectado entre el dispositivo y el primero de los focos a controlar, y una toma de corriente (que ya estará presente al ser utilizada por el foco de luz).

## 1.4.- Objetivos

El objetivo que se plantea en este proyecto, dentro de la línea explicada anteriormente, es el diseño e implementación de un sistema de control de iluminación con protocolo DMX de forma inalámbrica para poder instalarlo en una sala de estimulación multisensorial. Este sistema deberá integrarse dentro del protocolo de comunicación entre nodos ZigBee utilizado por Tecnodiscap, basado en un perfil privado ZigBee (al que en el resto de la memoria, por simplificar la lectura, nos referiremos como “Perfil ZigBee-Tecnodiscap”), y respetar la ontología OSGi4Aml, para ser totalmente interoperable y poder ser controlado con cualquier tipo de sensor de los desarrollados por el grupo.

Para ello se definen los siguientes objetivos parciales:

- Estudio inicial del estado del arte del DMX inalámbrico y las salas de estimulación multisensorial.
- Aprendizaje del protocolo DMX y sus posibilidades.
- Aprendizaje, diseño y desarrollo de un sistema de comunicación ZigBee.
- Aprendizaje y programación de un sistema de microcontrolador tipo PSoC para la gestión de los protocolos ZigBee y DMX.
- Elección y análisis de características del dispositivo de iluminación con control DMX apropiado para el sistema.
- Diseño del puente capaz de transformar la información recibida por ZigBee en una señal DMX para el dispositivo final:
  - Esquemáticos
  - Elección de componentes y montaje de prototipo en placa de inserción
  - Aprendizaje y diseño de la PCB
- Estudio del driver ZigBee del grupo Tecnodiscap e integración en él del dispositivo desarrollado.
- Aprendizaje e integración en OSGi con OSGi4Aml.
- Prueba de la solución final con un sensor desarrollado por el grupo, diseñando y desarrollando el software necesario.

## 1.5.- Estructura de la memoria

La memoria se ha estructurado en 6 capítulos, al final de los cuales se añaden 6 anexos que contribuyen a detallar aspectos que en la memoria, por falta de espacio, no se pueden tratar tan extensamente.

El contenido de cada capítulo es el siguiente:

- En el primer capítulo se da una visión general el proyecto, situándolo en el contexto de trabajo y especificando los objetivos y pasos que se han ido realizando.
- En el segundo capítulo se describe brevemente el estado del arte del DMX inalámbrico y las salas de estimulación multisensorial.
- En el tercer capítulo se profundiza en las especificaciones que definen las características que tendrá el sistema diseñado y se da una visión general del mismo.
- En el cuarto capítulo se da una visión general del protocolo DMX y se explica detalladamente todo el proceso de diseño e implementación hardware del sistema, así como la programación del firmware.
- En el quinto capítulo se expone el desarrollo middleware realizado para integrar el sistema en el driver ZigBee y en OSGi, así como el diseño software y la colaboración con alumnos y profesores de Diseño Industrial.
- En el sexto capítulo se realiza un análisis de los objetivos cumplidos, las conclusiones obtenidas y las líneas de trabajo futuras con el sistema diseñado.

En cuanto a los anexos:

- En el primer anexo se realiza una explicación detallada del concepto *snoezelen* y las salas de estimulación multisensorial.
- En el segundo anexo se muestran las principales características de PSoC.
- En el tercer anexo se detallan las características del protocolo DMX.

- En el cuarto anexo se explica el protocolo inalámbrico ZigBee.
- En el quinto anexo se presentan las imágenes de los diseños de la PCB y fotografías del montaje físico del sistema.
- En el sexto anexo se presenta el presupuesto del hardware diseñado.

## 1.6.- Distribución temporal del desarrollo del proyecto

A continuación [Fig. 1.1] se muestra la distribución temporal del proyecto con un diagrama de Gantt. Las tareas que se presentan se corresponden con los objetivos mencionados anteriormente.

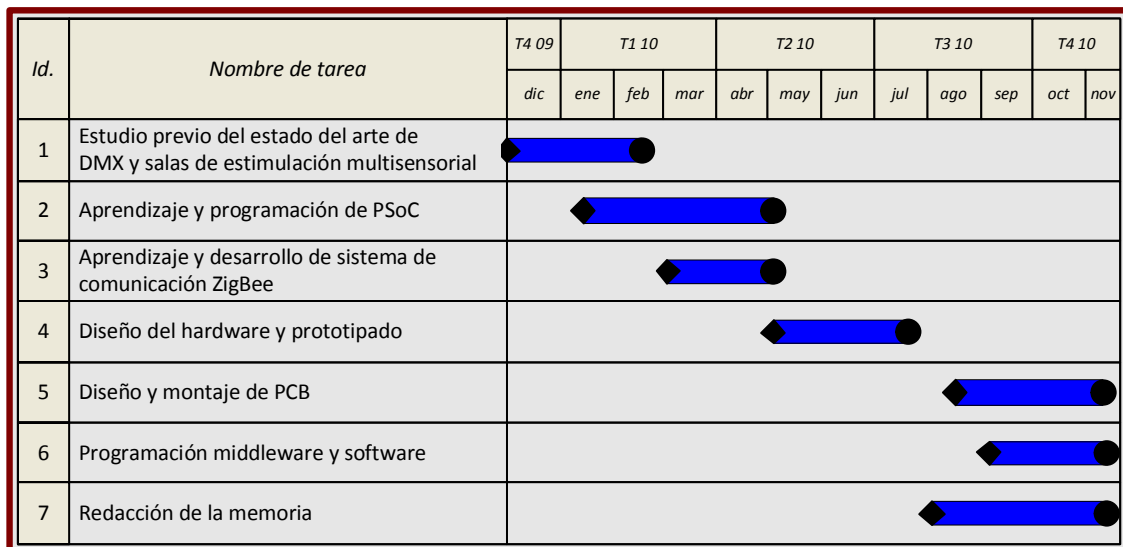


Fig. 1.1.- Distribución temporal del desarrollo del proyecto



## CAPITULO 2

### Estado del arte

El estudio detallado del estado del arte de las salas de estimulación multisensorial se ha planteado ya en el capítulo anterior, y se ha realizado una profundización en el mismo en el Anexo A, por lo que aquí nos centraremos en el estudio del control DMX inalámbrico.

Antes de diseñar nuestro sistema, se contactó con profesionales del sector del sonido y la iluminación de espectáculos, para tener una información concreta de la situación actual del control de dispositivos con protocolo DMX de mano de la gente que está trabajando con ello. Con esta información se obtuvo una buena visión de los sistemas disponibles en la actualidad.

A continuación pasamos a ver algunos de los más representativos, de entre todos los que se pueden encontrar en el mercado:

- **ENTTEC OPEN DMX USB**

Este dispositivo de la empresa Enttec [3] es un interfaz de puerto USB a conector XLR-5 [Fig. 2.1] con protocolo DMX, para poder sacar una línea de control DMX desde un ordenador y poder gestionar la iluminación de un espectáculo con un software sencillo y adaptado a las necesidades concretas. Su característica más interesante es que posee licencia GPL, es decir, su *hardware* es público y libre.

No se trata de un sistema inalámbrico como el diseñado en este proyecto, pero ha resultado muy útil su estudio, debido a que se consiguió uno por medio de un profesional del sector y se pudieron caracterizar las señales enviadas para el control de luces, obteniendo así unos datos reales de temporización de las tramas DMX y niveles de salida soportados.

A modo orientativo, el precio de este dispositivo es de 55 euros más 10 euros del adaptador XLR-5 a XLR-3, debiendo añadirle posteriormente, de querer ser utilizado, todo el hardware necesario para la funcionalidad inalámbrica.



Fig.2.1.- Interfaz USB-DMX de la empresa Enttec

- **CHAUVET D-Fi 2.0**

[4] Es uno de los sistemas profesionales más económicos del mercado, alrededor de los 200 euros cada elemento (siendo necesarios dos, uno configurado como emisor y el otro como receptor [Fig. 2.2], en total unos 400 euros) Este dato ya nos da una idea de la imposibilidad de integrar un sistema comercial en las salas de estimulación multisensorial, debido a que no se pueden permitir estos costes para el simple control de un dispositivo del aula.

Este sistema utiliza un protocolo propietario y trabaja en la banda UHF, pudiendo escoger la frecuencia de trabajo



Fig.2.2.- Sistema de control DMX inalámbrico comercializado por la empresa Chauvet

- **ELATION EWDMX SYSTEM**

Se presenta este sistema DMX inalámbrico [Fig. 2.3] como ejemplo típico de material profesional, cuyo precio no se adapta a nuestra capacidad y cuyas características técnicas, pensadas para espectáculos en directo, sobrepasan las necesidades de las aulas.

El sistema de *Elation Professional* [5] posibilita las comunicaciones a distancias de hasta 900 metros y permite controlar 16 universos DMX (un total de 8192 canales), magnitudes que no se corresponden con el uso que se hará del sistema en las salas.

Trabaja como una W-LAN en la banda de 2,45 GHz y utiliza técnicas de seguridad como FHSS (*Frequency Hopping Spread Spectrum*) para evitar interferencias con otras redes.

Asimismo, el precio del conjunto (emisor + receptor) ronda los 800-900 euros, debido a las características antes mencionadas (y manteniéndose todavía en el rango medio de precios en esta clase de sistemas)



Fig.2.3.- Sistema de control DMX inalámbrico comercializado por la empresa Elation Professional

- **HMBTEC DMX WiFi CONTROLLER**

Otra alternativa que encontramos en el mercado es la que se presenta en este dispositivo de la empresa HMBTEC [6] [Fig. 2.4]. La peculiaridad de este dispositivo es que está pensado para ser utilizado con una aplicación de dispositivos táctiles de la empresa Apple (iPhone, iPod y iPad) Esto no sería adecuado para nuestra implementación debido al incremento de precio que supondría depender de dispositivos comerciales de una sola marca (el software necesario debería de ser el vendido por la propia empresa).

En cuanto a especificaciones técnicas, trabaja bajo como una W-LAN WiFi y sólo nos permite controlar 100 canales DMX, limitándonos el tamaño del universo (que puede ser de hasta 512).



Fig. 2.4.- Sistema de control DMX inalámbrico comercializado por la empresa HMBTEC

## CAPITULO 3

# Especificaciones del sistema

### 3.1.- Esquema global del sistema

Para concretar lo anteriormente explicado, se presentan los siguientes esquemas. En el primero de ellos [Fig.3.1] observamos un diagrama físico que nos da una idea general del sistema completo y sus bloques funcionales. En el ordenador central instalaremos el software de control del sistema. Esta aplicación (que se presentará en el capítulo 5) establecerá la comunicación con todos los dispositivos de la sala de estimulación y los podremos configurar al inicio de la sesión. Particularizando a nuestro dispositivo, se observa en la figura que la comunicación se hará mediante protocolo ZigBee, teniendo un coordinador en el ordenador de control y un nodo en el dispositivo DMX.

Tras el nodo de comunicaciones, se situará el hardware diseñado para traducir las instrucciones recibidas vía ZigBee en datos con protocolo DMX, conectando a la salida del conjunto el dispositivo a controlar.

En el segundo esquema [Fig.3.2] se ilustra la pila de protocolos que se va a seguir en la programación firmware y middleware. La comunicación del firmware del microcontrolador con el sistema se enmarcará en el Perfil ZigBee-Tecnodiscap. Respetando este perfil, integraremos el dispositivo en el driver ZigBee utilizado por el grupo, que creará el dispositivo OSGi4Aml y publicará el servicio correspondiente en la capa OSGi.

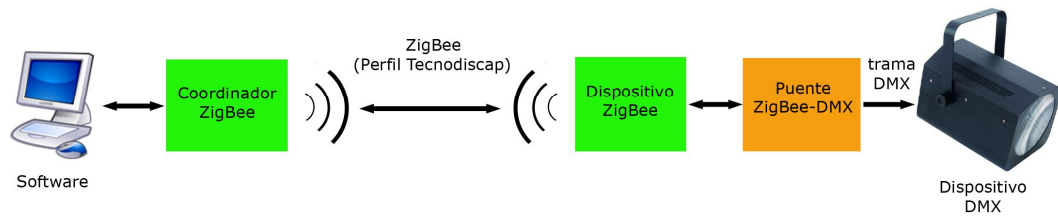


Fig. 3.1.- Diagrama físico del sistema completo

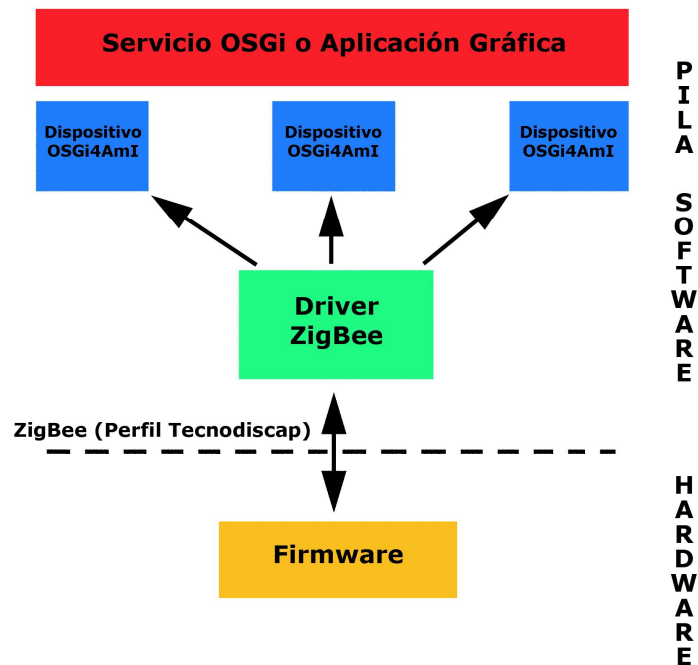


Fig. 3.2.- Esquema ilustrativo de la pila software del sistema y su comunicación con el hardware

### 3.2.- Necesidades del usuario y condicionantes

Teniendo en cuenta lo expuesto hasta ahora, se puede presentar una lista de los condicionantes que se nos presentan para el diseño del sistema, relacionándolos con las características principales que vendrán impuestas por éstos:

- **Sistema de bajo coste:** los colegios no poseen un gran presupuesto para el material y los diseños al no ser empresas al uso. Para conseguirlo optamos por:
  - **Dispositivos con protocolo DMX**, ya que al ser el estándar utilizado en iluminación y escenografía profesional, se encuentra una gran oferta disponible en el mercado, lo que abarata mucho el precio frente al material específico diseñado por la empresas especializadas para las salas.
  - **PSoC**. Con esta tecnología reduciremos la cantidad de hardware adicional en la PCB y su coste final. Sus principales características y ventajas se pueden consultar en el *Anexo B*, pero a modo de idea general podemos reseñar que se trata de un dispositivo que integra varios bloques funcionales en el propio chip del microcontrolador.
  - **Protocolo ZigBee**. Sus características cumplen perfectamente las necesidades del sistema y es más barato y sencillo que otras alternativas inalámbricas (consultar *Anexo D* para un estudio detallado de este protocolo)
- **Versatilidad:** el sistema diseñado será un controlador para todo tipo de dispositivos con protocolo DMX. Por lo tanto deberá de ser posible utilizarlo para manejar cualquier actuador con dicho protocolo que elijamos con cualquiera de los sensores que tiene desarrollados el grupo Tecnodiscap para este tipo de salas. Por ejemplo, para un foco en el que manejamos la aparición de los colores solamente, puede ser muy intuitivo un sensor basado en RFID, pero si en un futuro se cambiase el foco de luces por un foco motorizado, podría ser más intuitivo un sensor de inclinación que se correspondiese directamente con la posición del foco. Para lograr esto será necesario:
  - **Perfil ZigBee-Tecnodiscap**. Como indicamos en el apartado 1.4, es el protocolo de comunicación entre nodos ZigBee utilizado por Tecnodiscap. Su uso permitirá la integración del sistema en el driver ZigBee.

- **Driver Zigbee.** Integrándolo en el driver ZigBee utilizado en el grupo se consigue un dispositivo interoperable por cualquiera de los sensores desarrollados por Tecnodiscap.
- **OSGi4Aml.** Respetando la ontología OSGi4Aml conseguiremos que el sistema pueda ser utilizado por cualquier software (siempre que esté integrado en el framework OSGi) para el que sea transparente y sólo se centre en sus funcionalidades. Además, OSGi nos permitirá actualizar o modificar módulos en un futuro sin necesidad de actualizar todo el software ni detener su funcionamiento.
- **Aplicación de interconexiones.** Deberemos de diseñar una aplicación sencilla e intuitiva que nos permita configurar la sala como deseemos al comienzo de cada sesión asignando a cada controlador que tengamos disponible el actuador que se desee, y las características concretas que queramos relacionar entre ellos.



# **CAPITULO 4**

## **Desarrollo Hardware**

### **4.1.- Protocolo DMX**

En el *Anexo C* se ha realizado una explicación detallada de las características técnicas del protocolo DMX, pero a modo de introducción diremos que se trata del protocolo utilizado a nivel profesional en la iluminación y escenografía. Permite controlar todo tipo de dispositivos, desde focos de luz (tanto estáticos como motorizados) hasta elementos de efectos como máquinas de humo, lázers...

Nos ofrece una comunicación robusta frente a ruido (debido a que se controla con señal balanceada) y, gracias a que, como hemos comentado, es el estándar en el mundo profesional, nos abre un enorme abanico de material disponible para las salas de estimulación multisensorial.

### **4.2.- Actuador: foco de iluminación controlado por DMX**

#### **4.2.1.- Características**

Lo primero que se decidió al elegir el dispositivo de iluminación fueron sus características, realizando un estudio de varios dispositivos que se adecuaban al presupuesto que se manejaba (hasta unos 150 euros) En este punto ya se observa la importancia del diseño del sistema para que se pueda utilizar material del mercado profesional, puesto que la versatilidad de los equipos en este rango de valores ya es superior a la de equipos diseñados especialmente para salas de estimulación multisensorial, de coste muy superior.

Después de realizar un estudio de las características principales de diversos focos de iluminación DMX, se decide optar por focos LED de colores, desechando la posibilidad de focos motorizados debido a su coste (aunque su inserción en un futuro, si se solicitase por el colegio, sería inmediata debido a su integración en el control DMX)

Finalmente se escogió el modelo *Eurolite LED FX-250 RGBW DMX Flower Effect*. [7]. Este dispositivo nos ofrece una proyección de 156 leds de 10mm (48 x rojo, 36 x verde, 36 x azul, 36 x blanco, en lo sucesivo RGBW), agrupados en 6 *clusters*, y 15 programas de animación lumínica diferentes. Permite el control de la velocidad de las animaciones, el parpadeo y la intensidad individual de cada color, permitiendo un gran abanico de posibilidades para su uso en las salas. Además posee un micrófono integrado para que la luz responda al sonido.

La tecnología LED nos asegura además un bajo consumo del dispositivo.

#### 4.2.2.- Funcionamiento

El dispositivo en cuestión se controla mediante 6 canales DMX, configurables en la posición que deseemos dentro del universo DMX (del 1 al 6, del 7 al 12, del 13 al 18...) Para la explicación de los canales, sin pérdida de generalidad, supondremos que está configurado para funcionar con los canales del 1 al 6.

El canal 1 controla la intensidad de luz del foco (todos los colores), siendo el valor 000 apagado y 255 máxima intensidad.

Tiene tres modos de funcionamiento, dependiendo de los valores que le demos al canal 2, que podemos ver en la tabla siguiente [Tabla 4.1]

Valor canal 2	Función
0x00 – 0x0A	Colores RGBW independientes asignados a los canales 3, 4, 5 y 6 respectivamente
0x0B – 0xEA	15 programas de animación prediseñados, lanzados según el valor concreto (prog1 de 0x0B a 0x18, prog2 de 0x19 a 0x27, prog3 de 0x28 a 0x36...)
0xEB – 0xFF	2 programas de control por sonido (prog1 de 0xEB a 0xF9 y prog2 de 0xFA a 0xFF)

Tabla 4.1.- Modos de funcionamiento del foco dependiendo del valor del canal 2

Así, la función de los canales 3, 4, 5 y 6 depende del valor del canal 2, y se resume en la siguiente tabla [Tabla 4.2]

Canal	Valor canal 2	Función
3	0x00 – 0x0A	Intensidad rojo (0x00 apagado, 0xFF máxima)
	0x0B – 0xFF	Velocidad animación (0x00 estático, 0xFF máxima)
4	0x00 – 0x0A	Intensidad verde (0x00 apagado, 0xFF máxima)
	0x0B – 0xFF	Velocidad parpadeo (0x00 sin parpadeo, 0xFF máxima)
5	0x00 – 0x0A	Intensidad azul (0x00 apagado, 0xFF máxima)
	0x0B – 0xFF	-
6	0x00 – 0x0A	Intensidad blanco (0x00 apagado, 0xFF máxima)
	0x0B – 0xFF	-

Tabla 4.2.- Función de los canales 3-6 dependiendo del valor del canal 2

## 4.3.- PCB del puente DMX – módulo ZigBee

### 4.3.1.- Características y diagramas de bloques

Para comenzar el diseño del sistema se estudian algunas notas de aplicación de PSoC [8] y se observan las soluciones adoptadas y los problemas surgidos en sistemas similares con este tipo de tecnología para tener suficiente información previa al enfrentarnos a nuestro caso.

El primer paso es identificar los módulos necesarios para nuestro sistema, sus funciones, y realizar con ello un diagrama de bloques del hardware a diseñar [Fig. 4.1]

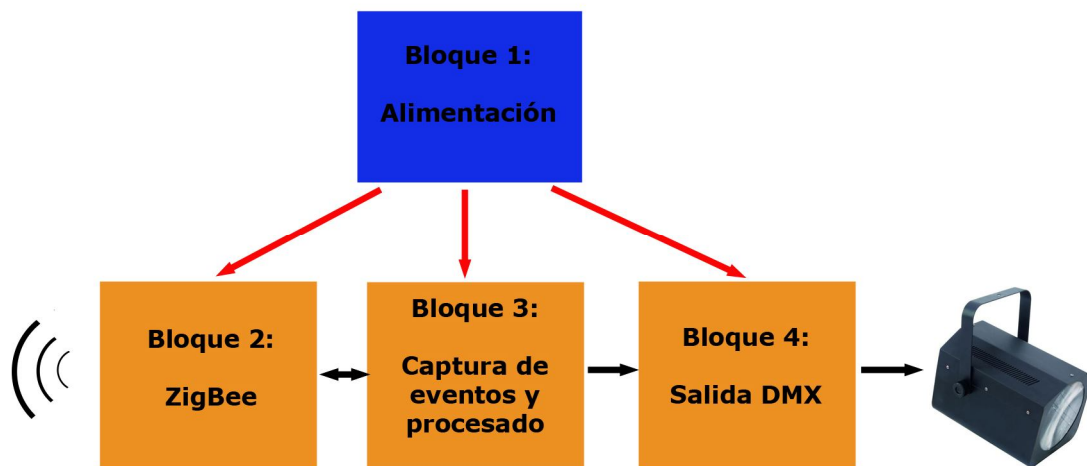


Fig.4.1.- Diagrama de bloques del sistema

- **Bloque 1: Alimentación.** Su función es alimentar a todo el sistema. La principal característica que debemos de tener en cuenta es que hemos de sacar dos alimentaciones distintas, debido a que el módulo ZigBee que utilizamos funciona con 3.3V y la salida al dispositivo DMX es una trama de datos entre 0V y 5V.

La toma de alimentación la realizaremos desde la red con un adaptador. Pondremos una resistencia de 0 $\Omega$  como fusible y un diodo Schottky como protección, y adecuaremos el nivel de voltaje a los dos necesarios en el sistema con reguladores lineales de bajo ruido MIC5209 [Fig.4.2] [9], en concreto MIC5209-5.0YS para la salida de 5V y MIC5209-3.0YS para la de 3.3V (alimentación del módulo ZigBee)

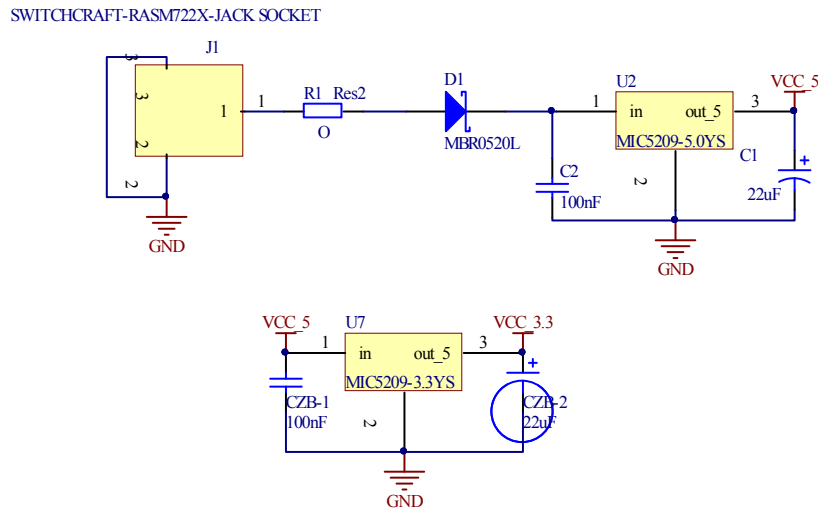


Fig.4.2.- Reguladores de voltaje MIC5209 para obtener los niveles deseados

- **Bloque 2: ZigBee.** Este bloque está compuesto por el módulo etrx2 [10] de Telegesis [11] Tiene como misión establecer las comunicaciones entre los diferentes elementos de la red. Para ello se comunica por medio de un puerto serie con el bloque de procesado por medio de comandos AT. Estos comandos, que están definidos por el fabricante [12], le comunican las acciones que tiene que realizar. En la figura [Fig.4.3] se puede ver el integrado ZigBee con sus conexiones.

Cabe destacar que la señal que sacamos del módulo UART de PSoC tiene niveles de 0V a 5V, por lo que hay que adaptar este nivel al aceptado por el módulo ZigBee, lo que realizaremos con un diodo Schottky y una resistencia. Cuando la UART mande un “0”, el diodo conducirá y tendremos el “0” en Zig\_RX, y cuando mande un “1”, cortaremos el diodo y tendremos los 3.3V en la salida [Fig.4.4]

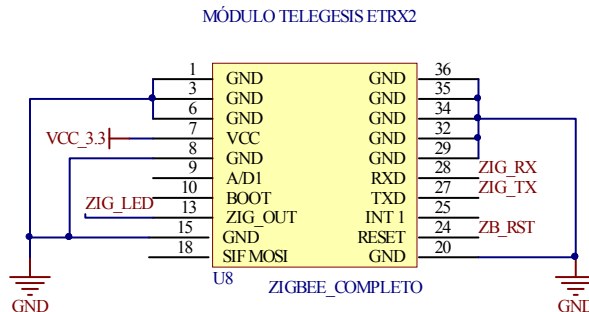


Fig. 4.3.- Módulo de comunicaciones ZigBee ETRX2 de Telegesis

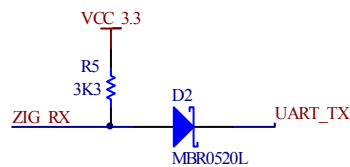


Fig. 4.4.- Adaptador de niveles de UART\_TX a Zig\_RX

- Bloque 3: Captura de eventos y procesamiento.** Gracias a la utilización de PSoC y sus características de configuración de bloques internos, podemos sintetizar en el chip todo el procesamiento de la señal de entrada y la captura de los eventos recibidos por el módulo ZigBee. En la figura [Fig. 4.5] se pueden observar todas las conexiones necesarias, así como las señales de entrada y salida

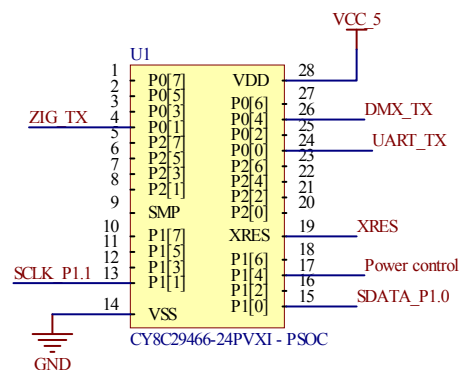


Fig. 4.5.- Señales de entrada y salida del PSoC

Para la carga del firmware en el chip y los futuros posibles cambios que haya que realizar en él, se colocará un conector en la placa para, mediante el *PSoC programmer*, se pueda acceder a su programación. La comunicación entre éste y el chip son las señales XRES, SDATA y SCLK que se observan en el esquema anterior.

La señal marcada como DMX\_TX es la encargada de enviar la trama con protocolo DMX, conformada en el firmware del chip, hacia el módulo de salida del sistema.

Asimismo, sacaremos una señal, indicada como Power control, que será la que permita o corte la alimentación del bloque de salida, permitiendo así que no se consuma cuando el foco esté en modo *OFF*.

Por último, las señales Zig\_TX y UART\_TX son las de transmisión y recepción de datos entre el PSoC y el módulo ZigBee.

- **Bloque 4: Salida DMX.** En este módulo es en el que se adecúa la trama DMX que formamos en el chip para sacarla con los niveles necesarios y, lo más importante, aislar la salida (tanto datos como alimentación) del resto del sistema.

Atendiendo a la alimentación del módulo de salida, la separaremos de la entrada mediante un convertor DC/DC aislado, en concreto el DCP010505BP DC/DC CONVERTER 1W 5V/5V de Texas Instruments [13]. El voltaje a la entrada lo controlaremos con un transistor MOSFET dependiente de la señal *Power control*, y su voltaje de salida lo utilizaremos para alimentar el módulo de salida del sistema [Fig.4.6]

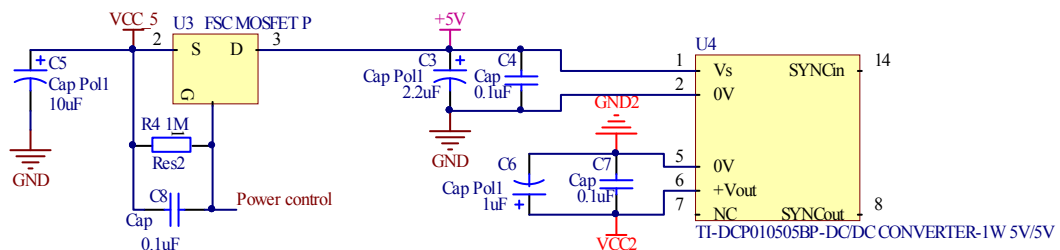


Fig. 4.6.- Aislamiento de la alimentación a la salida del sistema controlada desde el PSoC

En cuanto al módulo de salida, utilizaremos un optoacoplador para aislar la señal que le enviamos desde el PSoC de la trama que enviamos al dispositivo DMX. Dicha trama la simetrizaremos mediante un transceptor diferencial, el SN75LBC176AD DIFFERENTIAL BUS TRANSCEIVER de Texas Instruments [14], para conformar una señal balanceada que podamos sacar a través del conector XLR-3 [Fig. 4.7]

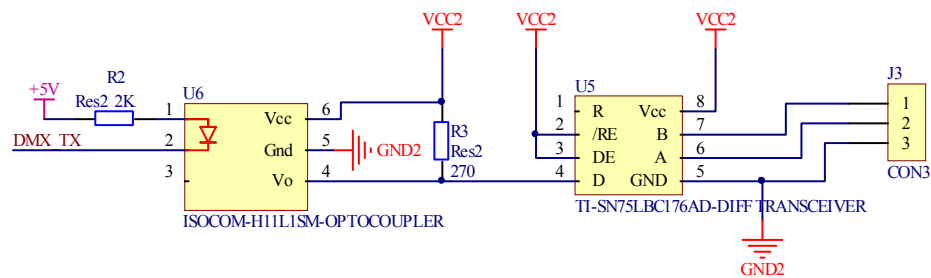


Fig. 4.7.- Salida del sistema hacia los dispositivos DMX

Observamos que todo el módulo de salida funciona con la alimentación Vcc2 referenciada a Gnd2, totalmente aislada del resto del sistema, lo que nos da una mayor seguridad y protección para las partes más sensibles.

#### 4.3.2.- Diseño de la PCB

Tras tener todo el hardware especificado, se procedió a montarlo en placa de inserción, comprobando su funcionamiento con el foco con una programación firmware a alto nivel, retocando lo que se estimó oportuno para su realización final.

Tras esto se diseñó la placa de circuito impreso, teniendo en cuenta para su diseño la separación física de tierras y alimentaciones entre el módulo de salida y el resto del sistema y dejando la unión con el conector XLR-3 mediante una ficha de tres contactos, para montar el prototipo dentro de una caja protectora pero dejar el diseño abierto a su instalación en otro tipo de contenedor en un futuro.

Tanto los esquemas de la PCB como las imágenes de ésta y la caja completamente montada, se pueden observar en el Anexo E.



## 4.4.- Firmware del puente DMX – módulo ZigBee

### 4.4.1.- Funcionamiento general

Para el control del sistema, como se ha comentado, se ha utilizado PSoC, en concreto el modelo CY8C29466-24PVXI [15]. Se encargará de recibir los comandos y datos procedentes del ZigBee y actualizar la trama DMX con los valores que le lleguen, así como construir dicha trama con las especificaciones temporales del protocolo DMX.

El programa principal (*main*) comenzará con una serie de instrucciones hacia el módulo ZigBee mediante comandos AT. Estas instrucciones son las encargadas de conformar la red ZigBee al inicio de cada utilización, y son las siguientes:

- **AT+DASSL** (*Dissasociate Local Device From PAN*): Desasociamos el módulo ZigBee de la red a la que pertenecía, asegurándonos de así de que no estaba asociado a otra red diferente a la nuestra.
- **AT+JN** (*Join Network*): Unimos el dispositivo a la red creada por el coordinador presente en la sala actual.
- **AT+SSINK** (*Search For a Sink*): Buscamos la dirección del coordinador de la red para poder comunicarnos con él por medio de mensajes SCASST.
- **AT+SCASST: Tecnodiscap DMX ZigBee Device** (*Transmit Data to the Sink*): Enviamos un mensaje de “presentación” al coordinador.

Para esta comunicación con el dispositivo ZigBee configuramos los siguientes módulos del PSoC:

- **UART**: este módulo nos permite crear una comunicación serie con el ZigBee (configurada a 19200 bps). La lectura de este puerto serie la realizaremos a bajo nivel, programando una rutina de servicio que lea a nivel de byte los datos recibidos y así mantener total control sobre la recepción de datos.
- **Sleep Timer**: es un reloj que utilizaremos para guardar los tiempos de espera entre comandos de configuración inicial del módulo ZigBee (del orden de segundos).

Tras la configuración de la comunicación ZigBee con el coordinador, entraremos en un bucle infinito que forma la trama DMX y la envía continuamente hacia el dispositivo, ya que es un requisito de este protocolo el refresco continuo de los valores de todos los canales. Para conformar dicha trama, configuraremos los siguientes módulos:

- **TX8:** es un puerto serie de 8 bits, por el que enviaremos la trama DMX hacia el bloque de salida del sistema.
- **PWM8:** gracias a este módulo formaremos el escalón del *break* al principio de cada trama. Programaremos una rutina de servicio tal que saque la señal del PWM por el pin de datos y, al terminar el *break*, conmute dicho pin de salida al TX8 para sacar los valores de todos los canales.

Además, crearemos una sencilla función de *delay* para los tiempos de espera del orden de  $\mu s$ , tales como los espacios entre canales o el tiempo entre tramas.

En cada iteración del bucle principal, guardaremos un tiempo de espera correspondiente al tiempo entre tramas, enviaremos la señal de *break* y, seguidamente, entraremos en otro bucle de 512 iteraciones en cada una de las cuales se enviará el valor del canal correspondiente.

En la siguiente figura [Fig. 4.8] se ve gráficamente el diagrama de flujo del proceso que se lleva a cabo en el firmware, observándose claramente las funciones desarrolladas por la rutina de servicio del puerto serie de recepción de datos.

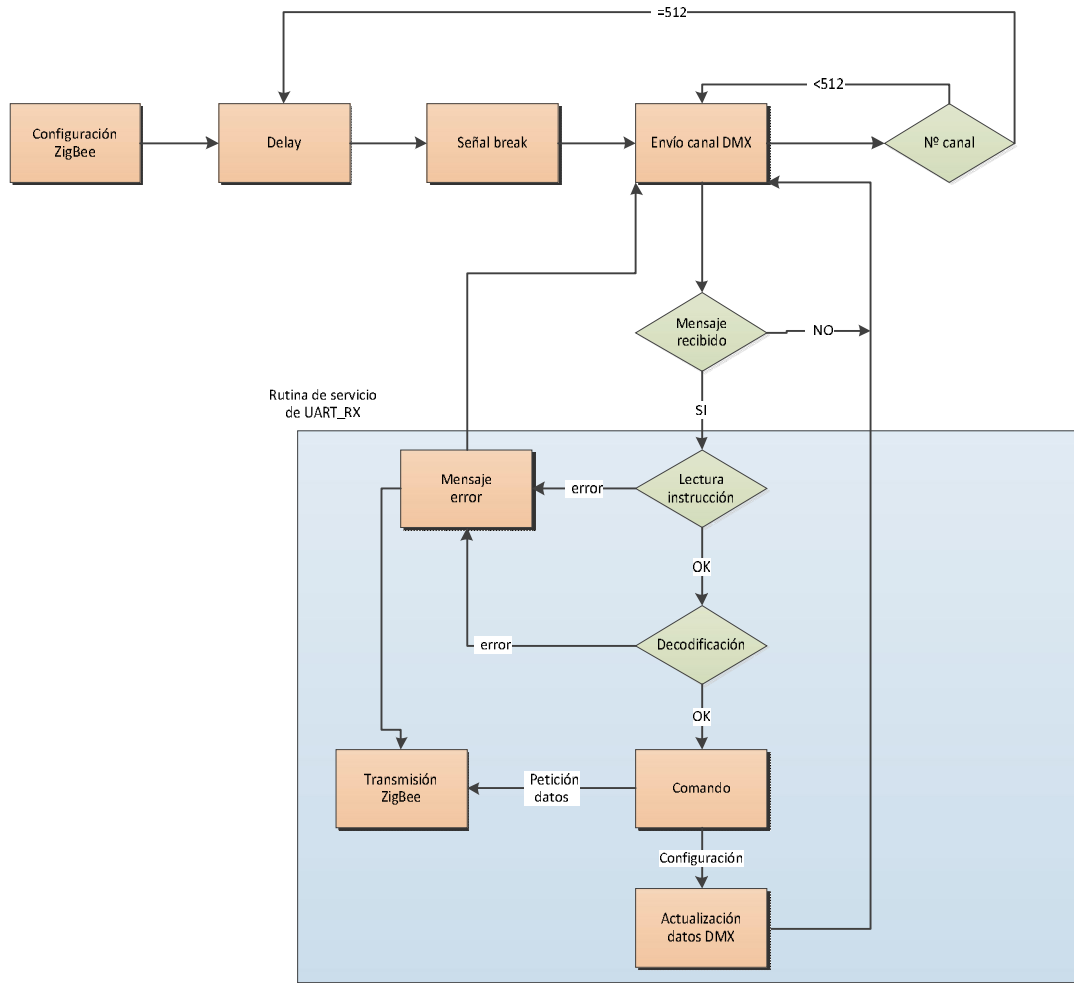


Fig. 4.8.- Diagrama de flujo del firmware

#### 4.4.2.- Perfil ZigBee-Tecnodiscap

Tras varias comprobaciones del sistema con la recepción de órdenes a alto nivel, se pasa a integrar las instrucciones en el Perfil ZigBee-Tecnodiscap.

El Módulo de comunicaciones ZigBee de Telegesis usado en el desarrollo permite un envío de mensajes entre los diferentes nodos de la red; a la forma de estructurar la información que va dentro de estos mensajes la hemos llamado Perfil ZigBee-Tecnodiscap. El Perfil ZigBee-Tecnodiscap tiene su origen en el proyecto Europeo MonAml [16]. Su definición fue liderada por la Universidad de Zaragoza y han participado diferentes socios: Telefónica I+D, Trialog y la Universidad de Passau.

Al hilo de lo explicado en el anexo dedicado a la tecnología ZigBee, el Perfil ZigBee-Tecnodiscap, siguiendo las directrices de la ZigBee Alliance, define diferentes «endpoints»,

«clusters», comandos y eventos. A continuación vamos a hacer una breve explicación de cada uno de estos conceptos.

Según el Perfil ZigBee-Tecnodiscap, cada dispositivo puede implementar diferentes endpoints, cada endpoint representa una funcionalidad individual que implementa el dispositivo; así por ejemplo, un sensor domótico que es capaz de sensar luz y temperatura, implementaría como poco dos endpoints, uno representaría el sensado de luz y el otro el de temperatura. Como se verá más adelante, la separación de funcionalidades por endpoints permite el acceso individual a cada una de las funcionalidades que puede dar cada endpoint.

Dentro de cada endpoint se pueden tener implementados diferentes clusters, cada cluster especifica una funcionalidad concreta dentro de un endpoint. Siguiendo con el ejemplo anterior del dispositivo sensor de luz y temperatura, dentro del endpoint que representa al sensor de temperatura podría estar implementado un cluster que emitiera una alarma (evento) cuando la temperatura superase un umbral. Los eventos y comandos son los mensajes específicos de cada cluster.

En las siguientes tablas se presentan los comandos y eventos más importantes implementados. El endpoint 04A\_BASE\_DEVICE [Tabla 4.3] es obligatorio implementarlo en cualquier dispositivo, pues es el endpoint que se encarga de informar al software del resto de endpoints implementados en el dispositivo. El endpoint DMX\_ACTUATOR [Tabla 4.4] es el propio de la aplicación y el que da la funcionalidad al foco DMX

0x10: 04A_BASE_DEVICE		
CLUSTER	COMANDO/EVENTO	DESCRIPCION
DEVICE COMMANDS	0X00: GET_TYPE	Solicita información de los endpoints
	0X02: PING	Verifica conexión
DEVICE EVENTS	0x00: ACK	Confirmación
	0x01: NOT_SUPPORTED	Comando no soportado
	0x02: ERROR	Error en algún campo del comando
	0x03: DEV_TYPE	Información de los endpoints

Tabla 4.3.- Comandos y eventos del endpoint 04A\_BASE\_DEVICE

0x11: DMX_ACTUATOR		
CLUSTER	COMANDO/EVENTO	DESCRIPCION
DEVICE COMMANDS	0x02: PING	Verifica conexión
DEVICE EVENTS	0x00: ACK	Confirmación
ACTUATOR COMMANDS	0x00: SET_STATUS	Configura canales DMX del actuador
ACTUATOREVENTS	0x00: STATUS_CHANGED	Confirmación de configuración de canales
ACTUATOR GET STATUS COMMANDS	0x01: GET_STATUS	Pide el valor de todos los canales DMX
ACTUATOR GET STATUS EVENTS	0x00: ACTUATOR_STATUS	Devuelve el valor de todos los canales DMX
ACTUATOR CUSTOM COMMANDS	0x00: SET_CONFIG_DMX_ACTUATOR	Establece el estado del dispositivo(ON/OFF)
	0x01: GET_CONFIG_DMX_ACTUATOR	Pide el estado del dispositivo (ON/OFF)
ACTUATOR CUSTOM EVENTS	0x01: CONFIG_DMX_ACTUATOR	Devuelve el estado del dispositivo (ON/OFF)

Tabla 4.4.- Comandos y eventos del endpoint DMX\_ACTUATOR

Desde el punto de vista de la programación en el PSoC, se ha creado una función que se ejecuta cada vez que le llega un mensaje procedente del ZigBee. Dicha función procesa el comando recibido, extrayendo la información del campo de datos (explicado en detalle en el apartado siguiente de este mismo capítulo), y entra en una instrucción de selección que va estudiando, nivel a nivel, el *endpoint* destino, el *cluster* y el comando recibido.

Una vez identificado el comando, se ejecuta la instrucción indicada en él (por ejemplo la actualización de los canales) y se devuelven, en su caso, los datos pedidos. Con estos datos se vuelve a formar la estructura del paquete del protocolo y se envía, vía ZigBee, al software de control del sistema.

#### 4.4.3.- Estructura de mensajes

A continuación se va a describir la estructura de los mensajes que intercambian los nodos ZigBee que obedecen al Perfil ZigBee-Tecnodiscap. Destacar que hay dos tipos de mensaje: eventos y comandos.

##### Comandos

Lo primero es definir que es un comando. Se denomina comando a todo aquel mensaje que parte del ordenador de control o de la pasarela residencial y tiene como destino cualquier nodo de la red.

En la figura [Fig.4.9] se muestra la estructura que tiene un comando tipo en la red implementada. A continuación se explicará brevemente que significa cada una de las partes que conforman el comando.

8 Bytes	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	N Bytes
ADDR_TO	END_POINT	PROFILE_ID	CLUSTER_ID	LENGTH	COMMAND	DATA
HEADER				PAYLOAD		

Fig.4.9.- Estructura de un paquete de tipo Comando

ADDR\_TO es la dirección a la que va remitido el mensaje, en nuestro caso no es necesario implementarlo porque el módulo ZigBee que utilizamos ya lo integra automáticamente igual que el PROFILE\_ID que es el perfil del dispositivo. La parte denominada como END\_POINT es el identificador, hexadecimal, del endpoint al que va dirigido el mensaje. CLUSTER\_ID es la identificación del grupo de atributos en la que se encuadra el mensaje. Estos cuatro componentes del mensaje conforman la parte denominada como HEADER, esta parte es siempre fija, es decir su tamaño es siempre el mismo y no se puede obviar a la hora de mandar un comando.

El resto del mensaje es la parte denominada PAYLOAD. Esta parte es de tamaño variable aunque también tendrá que ir siempre en el mensaje. La primera de sus partes LENGTH es la que define el tamaño del comando que se le envía al nodo, pero solo se expresa la longitud del PAYLOAD, puesto que como ya se ha explicado es la única parte del mensaje con tamaño variable. La segunda de sus partes es la denominada como COMMAND que, como su propio

nombre indica, es la identificación del comando que se manda al nodo correspondiente. Y por último, la parte denominada DATA, es la de tamaño variable, puesto que su longitud variará dependiendo del comando que se mande (e incluso puede no aparecer si el comando no requiere parámetros). En esta parte del mensaje se mandan los parámetros necesarios para su interpretación.

## Eventos

Igual que en el caso anterior lo primero es definir qué se entiende por evento. Se denomina evento a todo aquel mensaje que parte de un nodo de la red y tiene como destino el ordenador de control o la pasarela residencial.

Como se puede ver en la siguiente figura [Fig.4.10] la estructura es la misma cambiando pocas partes, por lo que solo se explicarán aquellas partes que se diferencian con respecto a los comandos.

8 Bytes	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	N Bytes
ADDR_FROM	END_POINT	PROFILE_ID	CLUSTER_ID	LENGTH	EVENT	DATA
HEADER				PAYLOAD		

Fig.4.10.-Estructura de un paquete del tipo Evento

La primera de las diferencias es que en lugar de mandar el nodo de destino, mandamos el nodo de origen. Por eso la primera de las partes, que como en el caso anterior está en la capa transparente al módulo ZigBee (por lo que no hay que programarla) es la denominada ADDR\_FROM.

El segundo cambio es el parámetro EVENT siendo éste, como su propio nombre indica, el evento que se envía para notificar cualquier circunstancia ocurrida al coordinador de la red.

# CAPITULO 5

## Middleware y software

### 5.1.- Framework OSGi y ontología OSGi4Aml

#### 5.1.1.- OSGi

Open Services Gateway Initiative (OSGi) es un grupo de trabajo que surgió en Marzo de 1999, cuyo principal impulsor es Sun Microsystems. El resto de los miembros fundadores son: Alcatel, Cable and Wireless, Electricité de France, Enron Communications, Ericsson, IBM, Liberate Technologies, Lucent Technologies, Motorola, Nortel Networks, Oracle, Philips Electronics, Sybase and Toshiba. Actualmente tiene unos treinta socios.

Su objetivo es definir y promover un estándar abierto para permitir conectar los servicios ofrecidos en redes metropolitanas (*WAN*) a redes de área local (*LAN*) o domóticas (*LON*). Esto permitirá la conexión de la próxima generación de aparatos inteligentes que se puedan encontrar en un hogar (oficina) con los servicios externos a la casa (oficina) ofrecidos a través de Internet. De esta forma, Proveedores de Internet (*ISP*), operadores de red y fabricantes de equipos pueden ofrecer una amplia gama de servicios a los usuarios finales, utilizando todos la misma pasarela.

La pasarela de servicios (*Services Gateway*) es un servidor embebido que se inserta en la propia red para conectar la red Internet externa con los clientes internos. La *SG* se inserta entre la red de Proveedores de Servicios y la red domótica, de tal forma que se separa la topología interna y externa de ambas redes. Esta *SG* debe ser capaz de manejar tanto flujo de datos como multimedia.



OSGi aporta a Java lo que se conoce como "Modularidad dinámica". Esto quiere decir que podemos instalar, actualizar o eliminar componentes de software *"en caliente"*. Además es capaz de descubrir librerías o aplicaciones de forma dinámica y utilizarlos en nuestros módulos o aplicativos.

En resumen, lo que OSGi aporta a Java es por una parte la capacidad de instalar módulos (*bundles* OSGi) de una manera dinámica, permitiendo así encenderlos y apagarlos tanto local como remotamente. Por otra parte OSGi nos permite que cada modulo publique objetos software (servicios o dispositivos) que pueden ser usados por el resto de módulos instalados.

Vamos a poner un ejemplo ilustrativo aplicado a aulas de estimulación multisensorial: supongamos que hemos desarrollado dos dispositivos hardware con sus respectivos módulos OSGi que publican los objetos software correspondientes a cada dispositivo hardware. Supongamos que los dispositivos son un foco de luces con protocolo DMX tele-controlado por ZigBee (como el diseñado en este proyecto) y una pantalla táctil. También supondremos que hemos desarrollado otro módulo OSGi (servicio) que permite el control de las luces con la pantalla táctil. Vamos a suponer que un tiempo después desarrollamos un guante con lector RFID. La ventaja de OSGi es que no tendríamos que cambiar todo el software y ni siquiera sería necesario detener la ejecución del sistema. Nos bastaría con desarrollar tanto el módulo que publique el nuevo dispositivo como el servicio/modulo que le de funcionalidad (por ejemplo el control de las luces mediante la lectura de *tags*). Además, estos módulos podrían ser instalados de forma remota en la maquina presente en el colegio, y desde ese momento podrían ser usados en la sala

### **5.1.2.- OSGi4Aml**

OSGi4Aml es una ontología para integración de dispositivos es sistemas basados en inteligencia ambiental. Su nombre es debido a que se ha implementado como una colección de interfaces Java para su uso en desarrollo con tecnología OSGi.

Define una formalización de los dispositivos que podemos encontrar en un ambiente inteligente, de forma que puedan ser utilizados por las aplicaciones de forma transparente, independientemente del fabricante del dispositivo, y centrándose sólo en la funcionalidad que requieren las aplicaciones.

Se pretende proporcionar una representación virtual de los dispositivos presentes en el entorno físico, de forma que puedan ser utilizados por las aplicaciones del ambiente inteligente sin preocuparse de otra cosa que de la funcionalidad requerida del dispositivo.

No busca definir todas las características que puede tener un dispositivo, sino sólo aquellas que realmente son relevantes para las aplicaciones, dejando explícitamente a un lado detalles de funcionamiento y configuración de los dispositivos, que en la práctica no son relevantes a las aplicaciones.

No es extraño que un mismo dispositivo físico pueda ofrecer distintas funciones relacionadas en mayor o menor medida. En esos casos, no tendremos un dispositivo virtual equivalente al dispositivo real, sino que existirán varios dispositivos simples que ofrecen la funcionalidad completa del dispositivo original.

Además de considerar dispositivos particulares, OSGi4Aml contempla atributos y funcionalidades que puedan ser comunes a dispositivos diferentes, agrupándolos en *clusters*, de forma que una implementación particular de un dispositivo concreto pueda seleccionar aquellos *clusters* que pueda ofrecer.

OSGi4Aml propone las bases para definir formalmente un dispositivo, pero da la libertad para definir nuevos dispositivos, incluso individuales, combinando las funcionalidades contempladas por los distintos *clusters*.

La ontología OSGi4Aml propone una jerarquización en niveles de los dispositivos, aproximadamente como se ve en la figura [Fig. 5.1]

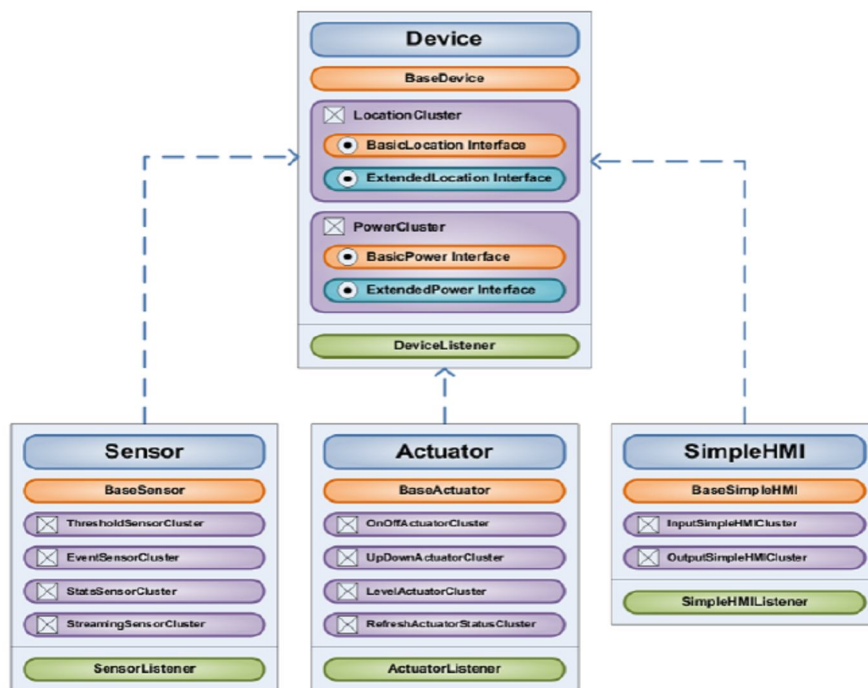


Fig. 5.1.- Jerarquización de los dispositivos según OSGi4Aml

Todos los dispositivos del entorno inteligente se ajustan a la definición del dispositivo básico BaseDevice, y en función del tipo de dispositivo concreto del que se trate, se ajustarán también a la definición de Sensor, Actuador o SimpleHMI, o de otros tipos que se pudieran definir.

OSGi4Aml permite abstraernos de la tecnología con la que estén implementados los dispositivos permitiendo intercambiar dispositivos con la misma funcionalidad aunque estén implementados en tecnologías diferentes.

La siguiente figura [Fig. 5.2] muestra la arquitectura del sistema de interacción para aulas de estimulación multisensorial. En ella puede verse cómo se utiliza OSGi y OSGi4Aml para combinar dos tecnologías diferentes: Bluetooth y Zigbee. En la figura se puede ver que existe un driver para cada una de las tecnologías que se ocupa de publicar los diferentes dispositivos físicos como dispositivos software en el framework. Gracias a OSGi4Aml los servicios en capas superiores pueden hacer uso de los dispositivos sin tener en cuenta la tecnología en la que están implementados. Por ejemplo, el servicio de interconexión de dispositivos puede utilizar tanto un acelerómetro basado en Zigbee como otro proveniente de tecnología bluetooth al publicarse ambos con el mismo interfaz OSGi4Aml.

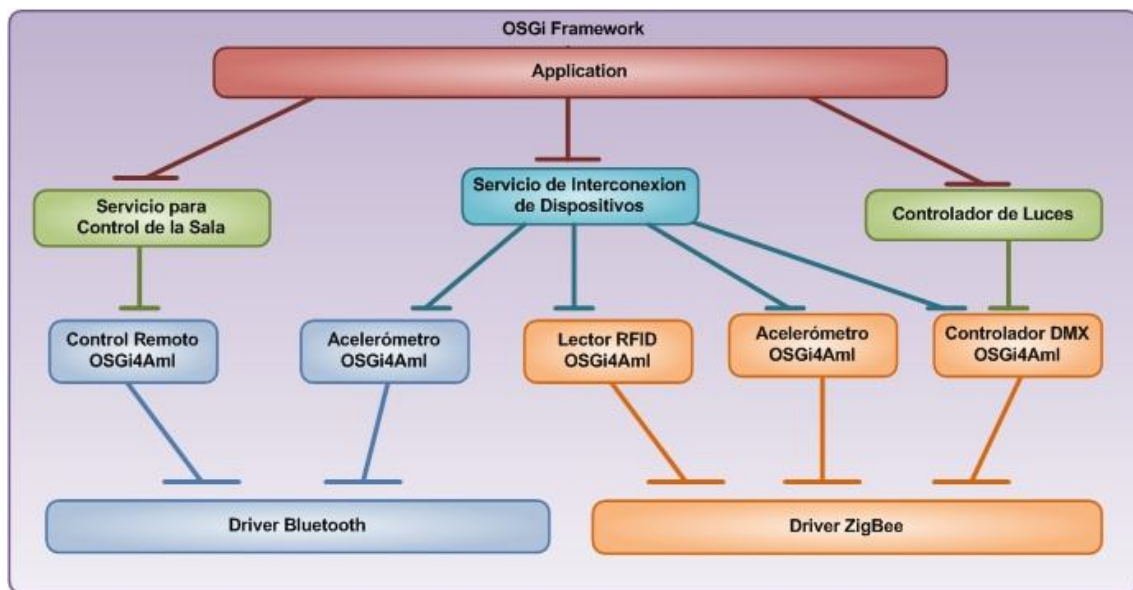


Fig. 5.2.- Ejemplo de arquitectura con OSGi y OSGi4Aml

### 5.1.3.- Relación entre OSGi4Aml y Perfil ZigBee-Tecnodiscap

Como hemos explicado en el apartado 4.4.2, un dispositivo ZigBee que obedece al Perfil ZigBee-Tecnodiscap implementa diferentes *endpoints* atendiendo a las funcionalidades individuales que ofrece, y a su vez dentro de cada *endpoint* implementa diversos *clusters* que dan funcionalidad específica a cada «sub-dispositivo».

Los diferentes *clusters* que puede implementar un *endpoint* están muy relacionados con la filosofía que persigue la ontología OSGi4Aml. Por una parte, todo dispositivo definido en OSGi4Aml hereda las propiedades y métodos del interfaz «Device». En este interfaz se describen métodos para la habilitación y des-habilitación del dispositivo, así como métodos para fijar y obtener información general del dispositivo.

Por otra parte, existe un cluster llamado «Device\_Cluster» definido en el Perfil ZigBee-Tecnodiscap que permite en el dispositivo físico realizar las operaciones de habilitación, recuperación de información... Puede verse que de esta manera queda enlazadas las operaciones que puede realizar el dispositivo físico con las que permite realizar el objeto software que lo controla en el middleware.

Para tratar de aclarar la relación entre Perfil ZigBee-Tecnodiscap y OSGi4Aml, vamos comparar los *clusters* implementados por nuestro dispositivo (ver *Tabla 4.4* del capítulo anterior), con los interfaces OSGi4Aml que implementarán el objeto software. De esta manera se podrá ver la estrecha relación que guardan unos con otros.

```
package es.unizar.tecnodiscap.osgi4aml.device.actuator;

import es.unizar.tecnodiscap.osgi4aml.device.Device;
import java.util.Map;

public interface Actuator extends Device {

    public static final String KEY_STATUS = "status";

    public int setActuatorStatus (Map status);

    public int registerActuatorListener (ActuatorListener listener);

    public int unregisterActuatorListener (ActuatorListener listener);

}
```

Fig. 5.3.- Interfaz OSGi4Aml Actuator

```

package es.unizar.tecnodiscap.osgi4ami.device.actuator;

import java.util.Date;
import java.util.Map;

public interface GetStatusActuatorCluster {

    public Map getActuatorStatus ();

    public Date getActuatorStatusTimestamp ();

    public int refreshActuatorStatus ();

}

```

Fig. 5.4.- Interfaz OSGi4Aml GetStatusActuatorCluster

Todo *endpoint* implementa el cluster DEVICE equivalente al interfaz “Device” de OSGi4Aml del que heredan todo los dispositivos. A su vez el *endpoint* DMX\_ACTUATOR implementa los *clusters* ACTUATOR, ACTUATOR\_GET\_STATUS y ACTUATOR\_CUSTOM, correspondientes a subtipos de dispositivos OSGi4Aml. A modo de ejemplo, podemos comparar la definición de los interfaces OSGi4Aml “Actuator” [Fig. 5.3] y “GetStatusActuatorCluster” [Fig. 5.4] con los métodos y eventos de los clusters ACTUATOR y ACTUATOR\_GET\_STATUS que vimos en la en capítulo anterior [Tabla. 4.4]. Puede verse una correspondencia directa de métodos.

Como resumen decir que OSGi4Aml es una ontología que permite la abstracción del hardware en la capa middleware y el Perfil ZigBee-Tecnodiscap es la forma en la que el middleware se comunica con el hardware, pero diseñados de una manera concordante permiten realizar comunicaciones eficientes debido a que cuando se requiere alguna información del hardware sólo se transmite la información mínima.

## 5.2.- Driver ZigBee

Tras la puesta en funcionamiento del sistema bajo Perfil ZigBee-Tecnodiscap, se pasa a integrarlo en el driver ZigBee utilizado por el grupo Tecnodiscap para todos sus dispositivos inalámbricos que utilizan esta tecnología. Este paso es de gran importancia, pues nos aporta:

- Autodetección del dispositivo ZigBee por parte del software
- Reconocimiento de los endpoints presentes
- Publicación en OSGi de los servicios correspondientes

La primera tarea que realiza el driver es escanear uno por uno todos los puertos serie activos del ordenador, hasta encontrar el que está conectado con un coordinador ZigBee. Una vez hallado, establece una comunicación con él y pasa a explorar la red que establece dicho coordinador, detectando todos los nodos ZigBee pertenecientes a ella.

Tras la detección de los nodos, el driver pregunta a cada uno de ellos por los *endpoints* que tienen conectados, y crea para cada uno un nuevo dispositivo que albergue las características establecidas en ese tipo de *endpoint*.

Una vez creados los dispositivos, publica en OSGi los servicios correspondientes a cada uno, y gracias a OSGi4Aml, el software que utilice estos servicios se abstrae de la tecnología que hay debajo y los utiliza de forma transparente.

En el siguiente diagrama [Fig.5.5] podemos observar el proceso desde la detección del nodo hasta la publicación del servicio en OSGi.

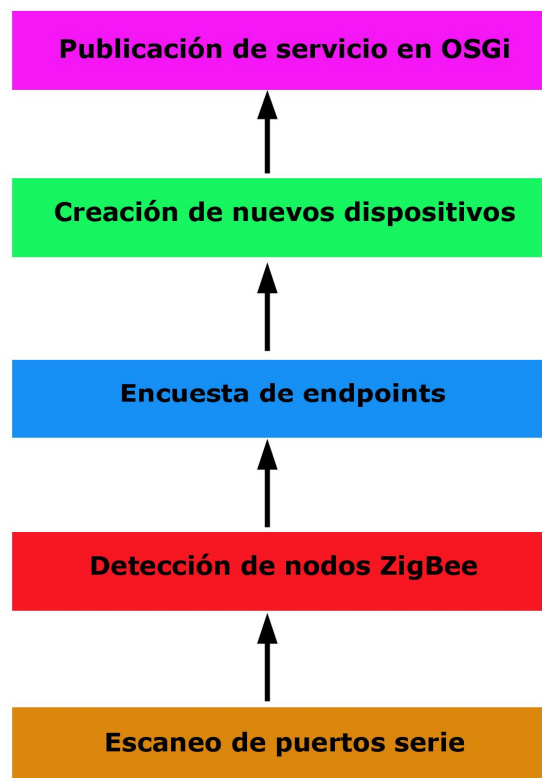


Fig.5.5.- Diagrama de funcionamiento del driver ZigBee

La labor principal que se ha realizado dentro de este driver ha sido la creación e integración del *endpoint* diseñado para nuestro sistema, que como vimos en el apartado 4.4.2 es el llamado DMX\_ACTUATOR.

Tras la implementación de este nuevo *endpoint*, se ha asociado a la creación del dispositivo correspondiente para su publicación en OSGi, quedando el sistema totalmente integrado en el driver ZigBee.

## **5.3.- Desarrollo software**

### **5.3.1.- Colaboración con alumnos y profesores de Diseño Industrial**

Al comienzo del curso 2009-2010 se mantuvieron contactos con el profesor Rubén Rebollar del Área de Proyectos del Departamento de Diseño y Fabricación para realizar una colaboración con los alumnos de la asignatura Interacción Usuario-Producto de la titulación de Ingeniería Técnica Industrial en Diseño Industrial.

El objetivo de la colaboración era el de obtener una serie de propuestas sobre cómo debería ser el interfaz de usuario de un futuro sistema que mejorara las actuales aulas de estimulación multisensorial, a la vez que los alumnos se enfrentaban a una situación de diseño más cercana a la realidad y que les permitía tratar con futuros usuarios de sus diseños.

Para comenzar con la colaboración primero se realizó una presentación en la que se expuso a los alumnos el concepto de aulas de estimulación multisensorial y se presentaron los principales elementos con los que se cuenta actualmente. Se les presentaron las principales dificultades a las que se enfrentan los profesores de educación especial (principalmente falta de interactividad entre elementos) y se les propusieron posibles elementos tecnológicos a incorporar que el grupo Tecnodiscap considera factibles o fácilmente desarrollables.

Una vez introducidos en el tema, se convocó una sesión de *brainstorming* con el objetivo de que los alumnos pensasen posibles interacciones entre los elementos y las tecnologías que se les había presentado. De esa sesión de *brainstorming* se fijaron una serie de elementos que debían ser controlados por un hipotético sistema informático/electrónico. El trabajo de asignatura de estos alumnos consistió en diseñar por grupos un interfaz de usuario para este hipotético sistema.

Por concretar un poco más las especificaciones de ese interfaz, el trabajo consistía en diseñar el interfaz gráfico de la parte de planificación de sesiones para cada niño, junto con el diseño

de un interfaz para controlar el sistema dentro de la sala una vez se está trabajando con un niño en concreto.

Antes de comenzar con el diseño de los interfaces se realizó una visita al colegio de educación especial Alborada para ver de cerca una sala de estimulación multisensorial y poder ver cómo los profesores de educación especial desarrollan su trabajo.

Una vez realizadas unas primeras versiones de los interfaces de cada grupo realizó una prueba con usuarios reales, en concreto los profesores del colegio Alborada. Esta prueba sirvió para realizar ajustes en cuanto a funcionalidad y usabilidad de cada una de las propuestas.

Las propuestas de los alumnos en cuanto a la planificación de las sesiones giraron principalmente en torno a la planificación del uso de elementos dentro de un orden temporal, al estilo de un diagrama de Gantt.

Con respecto al interfaz de control dentro de la sala hubo menos unanimidad en la solución y la variedad de propuestas fue desde un control basado en dispositivos tipo PDA o teléfonos táctiles, pasando por controles basados en mando a distancia tipo Wii hasta controles por voz mediante dispositivos tipo *headset bluetooth*.

Al final del curso cada grupo realizó una presentación de su desarrollo así como una valoración de la experiencia de trabajo sobre un caso práctico real.

En general la experiencia fue muy positiva, resultando enriquecedora tanto para los alumnos como para el grupo Tecnodiscap. En el plano personal fue muy enriquecedor el enfrentarme a la preparación y exposición de una presentación frente a un grupo de alumnos, intentando aclararles las dudas sobre el tema.

### **5.3.2.- Aplicación de desarrollo para el control del foco**

Además de la aplicación para la interconexión de dispositivos, y a modo de presentación de las características del foco de iluminación elegido para este proyecto, se ha diseñado un software de control que muestra las posibilidades del dispositivo en cuestión.

Se trata de una sencilla consola que permite enviar órdenes directamente al foco DMX desde el ordenador, y lanzar los programas prediseñados.

Si analizamos la imagen del interfaz [Fig. 5.6], se observa que la pantalla ofrece dos métodos de control: “Programas prediseñados” y “Control por colores”



El primero de ellos (“Programas prediseñados”) nos permite lanzar cualquiera de los 15 programas de iluminación automática que posee el foco, pudiendo controlar la velocidad de ejecución de la animación y la frecuencia de parpadeo de la luz. Todo esto se ha hecho de manera que resulte un manejo muy sencillo, para que simplemente pulsando un botón se ejecute cada una de las acciones.

El segundo de los métodos de control (“Control por colores”) nos permite controlar la intensidad de iluminación de cada uno de los 4 colores por separado, obteniendo una luz constante y sin animación (para crear un color ambiental, por ejemplo).

Como se ha comentado, este software tiene una clara orientación hacia la presentación de las características y posibilidades del foco escogido, pero se puede integrar fácilmente, si se deseara, en la aplicación de interacción de dispositivos como un menú de configuración del foco para cada sesión.

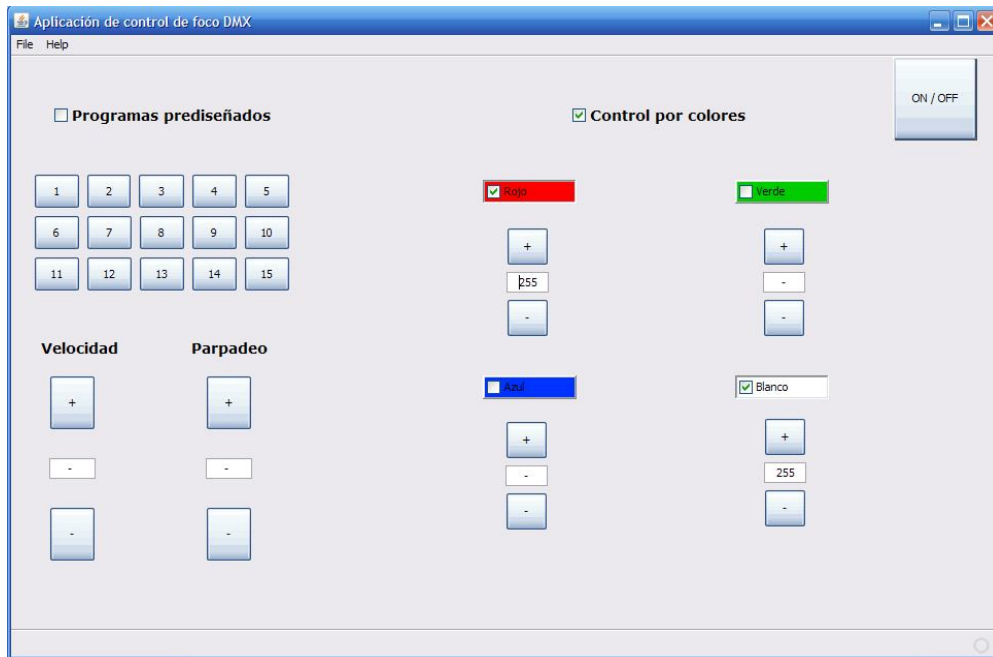


Fig. 5.6.- Interfaz de usuario de la aplicación para el control del foco

### 5.3.3.- Aplicación de interacción entre dispositivos

En apartados anteriores hemos visto como a través del framework OSGi y de la ontología OSGi4Aml hemos sentados las bases para la creación de un sistema que permita añadir a las aulas de estimulación multisensorial dispositivos capaces de interactuar entre ellos y con el usuario.

En esta sección vamos a mostrar una aplicación desarrollada para permitir a los profesores de educación especial configurar los diferentes dispositivos instalados en la sala para que interactúen unos con otros. Esta idea surge de la colaboración realizada con alumnos de diseño industrial, que se ha expuesto en el apartado 5.3.1.

En la figura [Fig. 5.7] puede verse el interfaz de usuario de la aplicación. Como puede observarse se ha tratado de simplificar al máximo el interfaz, dado que los profesores de educación especial no tienen por qué estar habituados al manejo del ordenador.

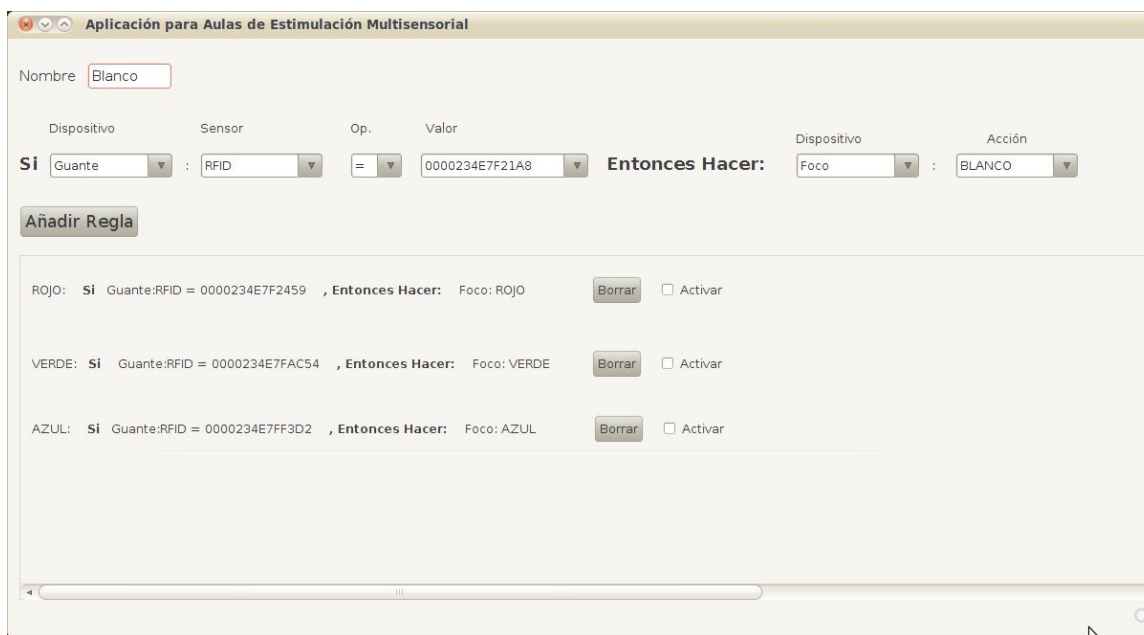


Fig. 5.7.- Interfaz de usuario de la aplicación de interconexión entre dispositivos

Se ha implementado un sistema intuitivo en el que el profesor sólo tiene que relacionar un dispositivo tipo «sensor» con un dispositivo tipo «actuador». Se busca que el profesional relacione una acción del usuario con una reacción del sistema. Por ejemplo, si el usuario toca o se acerca a algún objeto marcado con una etiqueta RFID concreta, deseamos que el sistema reaccione cambiando el color de la luz ambiental (configuración del interfaz presentada en la imagen Fig. 5.7).

Esto se consigue gracias una lista de reglas activas y un método para añadir nuevas reglas. Como puede verse en la imagen del interfaz [Fig. 5.7], en la parte inferior de la aplicación hay una lista de reglas que pueden estar activas o no, a elección del profesor. También es posible

añadir nuevas reglas y borrar las ya existentes. Para añadir nuevas reglas se utiliza la parte superior de la aplicación, donde pueden verse diferentes listas desplegables.

El profesor debería primero elegir el dispositivo sensor. En esa lista aparecerá un nombre amigable e intuitivo para cada dispositivo desarrollado que se vaya añadiendo al aula (por ejemplo “Guante” para un dispositivo de interacción basado en RFID que se coloca en la mano del usuario a modo de guante o “Mando” para utilizar el mando de la consola Wii como controlador)

Una vez elegido el dispositivo, hay que elegir la acción del usuario a detectar, ya que puede darse el caso de que se puedan detectar varias acciones del usuario con diferentes sensores colocados en el dispositivo de control (utilizando el ejemplo del “Guante”, queremos que cada uno de los sensores que tiene, como el lector RFID y el sensor de actividad, controle un efecto en la sala). Para concretar un poco más la acción a detectar se han añadido las dos listas desplegables siguientes: en la primera hay que elegir un operador lógico ( =, >, < ...) y en la siguiente hay que concretar el valor de comparación.

Por último, siguiendo con este ejemplo, sólo quedaría elegir el dispositivo sobre el que actuar y la acción que debe realizar dicho actuador. Por ejemplo, escogiendo “Foco” para activar las opciones del dispositivo DMX diseñado en este proyecto, podríamos señalar que se cambien las luces a un determinado color.

# **CAPITULO 6**

## **Conclusiones**

### **6.1.- Cumplimiento de objetivos**

Tras la finalización de este proyecto se ha conseguido tener funcionando un sistema de control de iluminación con protocolo DMX de forma inalámbrica para poder instalarlo en una sala de estimulación multisensorial.

Asimismo se ha integrado dicho sistema en los protocolos de comunicaciones utilizados en Tecnodiscap, quedando como resultado un sistema totalmente interoperable con cualquier sensor creado por el grupo.

Si analizamos los objetivos parciales:

- Se ha estudiado el protocolo DMX y se ha contactado con profesionales del sector para tener una visión cercana del mercado. Asimismo se ha contado con la experiencia de la gente del grupo en el ámbito de las salas de estimulación multisensorial para tener presentes en todo momento las características especiales de este tipo de proyectos y se ha colaborado con alumnos y profesores de Diseño Industrial para la programación del software de control.
- Se ha profundizado en la tecnología PSoC y, gracias a ella, se ha conseguido un sistema con un hardware muy reducido y sencillo.

- Se ha conseguido configurar una comunicación ZigBee y enmarcarla dentro del driver ZigBee utilizado por el grupo, quedando un sistema que puede funcionar en presencia de otros desarrollados sin interferir en su funcionamiento.
- Se ha diseñado y montado una PCB, y se ha preparado una caja protectora con los conectores necesarios, quedando un conjunto compacto y seguro.
- Se ha programado un software de desarrollo con el que se ha comprobado el correcto funcionamiento del sistema, y que además sirve como muestra de las características particulares del dispositivo DMX escogido para el proyecto.
- Se ha diseñado e implementado un software de control para la programación de las interconexiones entre dispositivos, que servirá de base para la puesta en funcionamiento final en un aula de estimulación multisensorial para permitir a los profesores configurar el sistema en cada sesión según sus necesidades. Se ha tenido en cuenta para ello el trabajo realizado en colaboración con los alumnos y profesores de Diseño Industrial.
- Se ha probado el sistema con un sensor desarrollado por Tecnodiscap y se ha verificado su funcionamiento utilizando el software indicado en el punto anterior.

En el plano personal, la realización de este PFC ha supuesto un gran aprendizaje, tanto de herramientas de diseño como de métodos de trabajo. Se ha estudiado y alcanzado cierto nivel de manejo de PSoC, ZigBee, Altium y Java. Además, el diseñar y montar una PCB ha permitido conocer aspectos técnicos como la soldadura y el mecanizado.

En resumen, se ha conseguido tener una visión global del proceso de diseño y creación de un dispositivo, trabajando en contacto con gente con una gran experiencia y conocimientos técnicos y obteniendo de ello un bagaje muy positivo.

Por todo lo comentado, se puede concluir que se han cumplido satisfactoriamente todos los objetivos planteados al inicio del proyecto.

## **6.2.- Desarrollos futuros**

El primer paso natural que se prevé tras este proyecto es la puesta en funcionamiento del sistema en una sala de estimulación multisensorial para ser evaluado por los usuarios. Tras unas sesiones de verificación y test, se modificará lo que se crea necesario y se instalará la versión definitiva para su utilización por parte de los profesionales en estas aulas.

Además, dada la generalidad con que se ha hecho el sistema de control DMX, teniendo en cuenta los 512 canales disponibles en el protocolo y no sólo los que utiliza el dispositivo utilizado, es directo el ampliar el número de dispositivos de iluminación controlados o la complejidad de los mismos, pudiendo mejorar con éstos (si los terapeutas lo creen así) los resultados obtenidos en los usuarios.

Por otro lado, al ser un diseño pensado desde el principio para estar acorde con todos los protocolos y métodos de trabajo del grupo Tecnodiscap, se presta a ser integrado en otros sistemas desarrollados, para finalmente lograr un gran abanico de sensores y actuadores interoperables que den a las salas de estimulación multisensorial unas características y posibilidades de las que actualmente carecen.

En cuanto al software de control de las interconexiones entre dispositivos, se irá ampliando en relación a los elementos que se vayan desarrollando en el grupo, adaptando las posibilidades de relación sensor-actuador a los diseños concretos que se realicen. Asimismo, el diseño del interfaz y los métodos de configuración de las sesiones se podrán modificar atendiendo a las indicaciones que se den por parte de los terapeutas en las sesiones de test, para que finalmente quede una aplicación con la que se sientan familiarizados a la hora de trabajar y les aporte las características deseadas.

